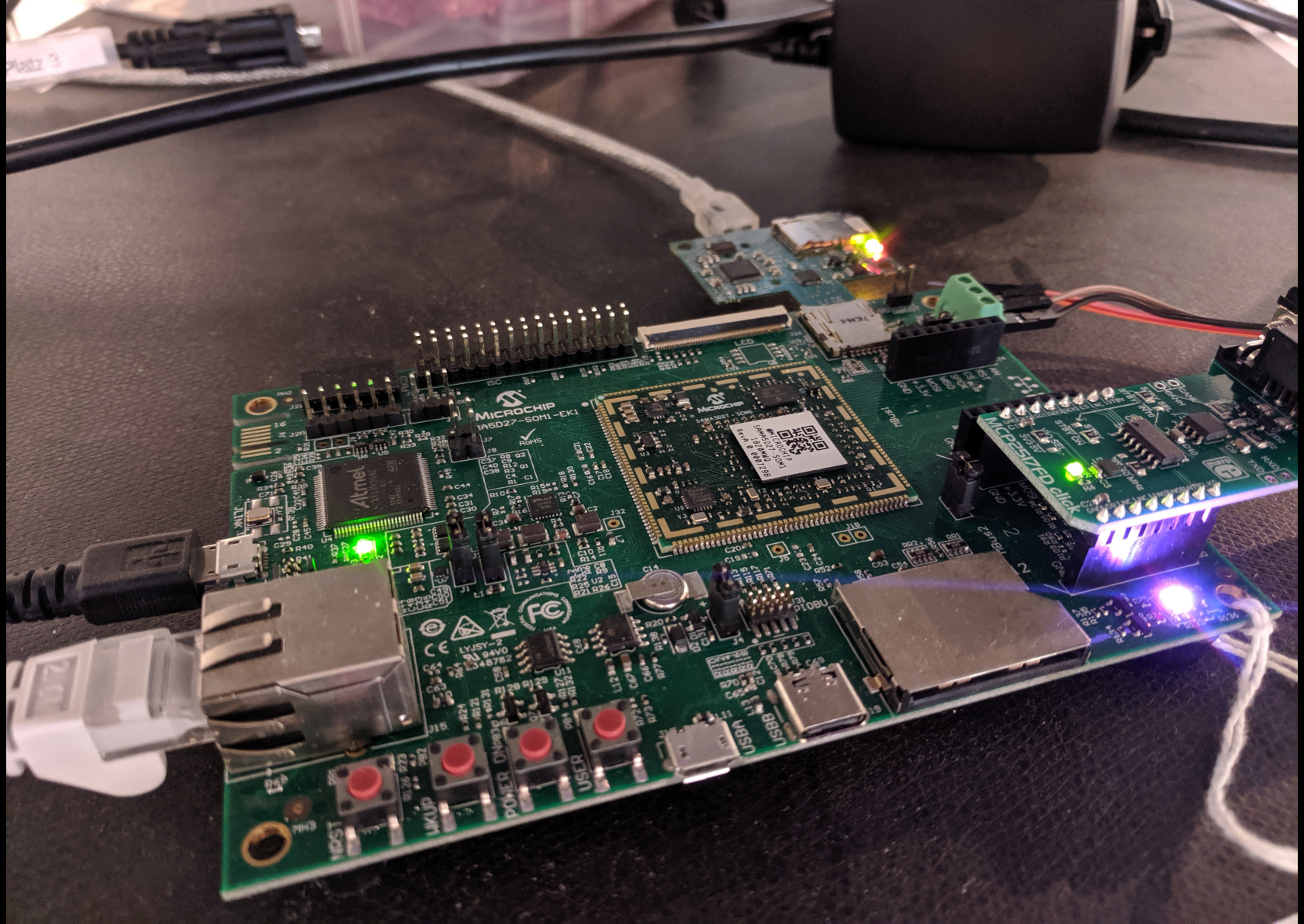
real-world examples

ATS 2019 - Lyon

Jan Lübbe – jlu@pengutronix.de
Rouven Czerwinski – rcz@pengutronix.de


Platz 3
MICROCHIP
SAMA5D27-SOM1-EK1
Atmel
MIKROE
mikroBUS
USBA
USBB
USER
WKUP
RSTC

Agenda

- Short overview of labgrid: scope, use-cases and architecture
- Example: OP-TEE on i.MX6
  - local and remote access
  - automated provisioning (“strategies”)
  - testing of: bootloader, OP-TEE init, TA usage under Linux
  - reservations
- controlling diverse hardware

Board Access @ Pengutronix

- Customer sends one board
  - “We want tests as well!”

  → Allow developer access and scheduled CI test runs
- Use existing infrastructure to ease dev tasks
  - Bootstraping device
  - Booting to Linux (do what’s necessary)

  → Reuse test information for board control
- Bootloader Testing

  → Always bootstrap the whole system

Tests and CLI access

- Pytest Interface for tests
  - Reuse pytest fixtures infrastructure
  - Reuse pytest hooks for collection, modification,...
- CLI interface for remote boards
  - Turn board on/off, open console, switch IO, show webcam video, ...
- Infrastructure to reserve and lock boards
  - Only one board for several devs, requiring coordination

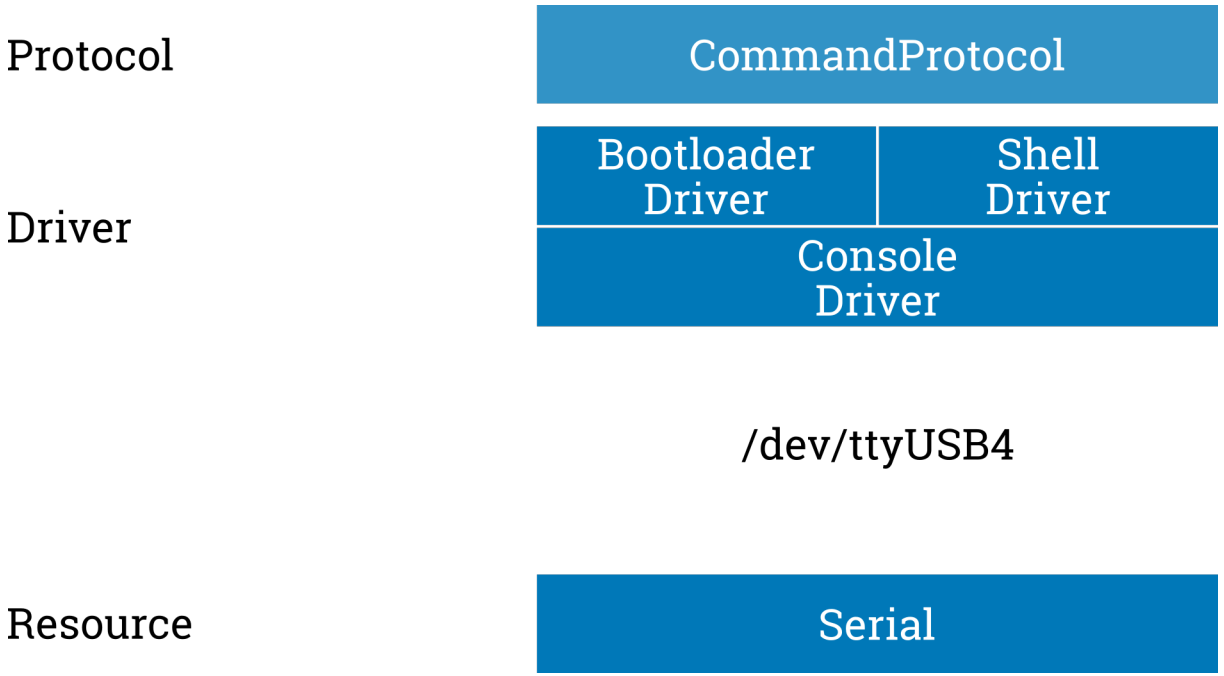
Remote Transparent Configuration

- Switching from local to remote should be easy
- Workflow
  - Bringup at desk
  - Put into lab
  - Replace hand-defined resources with RemotePlace
  - ~~???~~
  - Done!

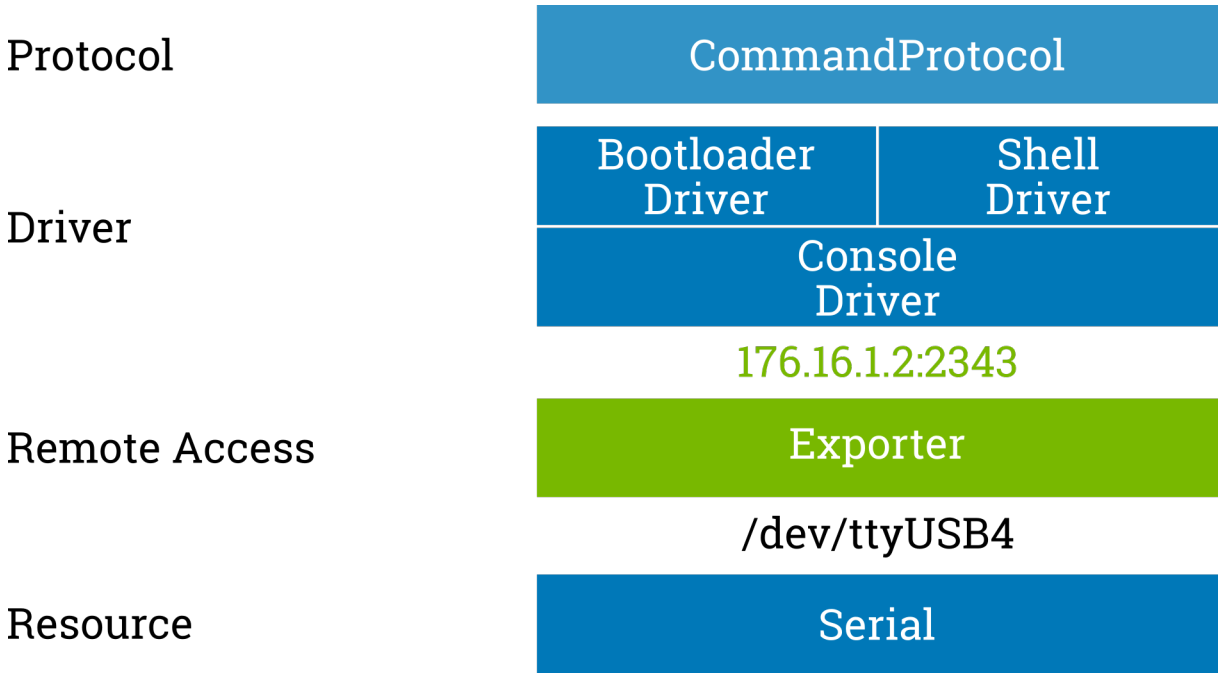
Try ~~Something Else~~ Less

- no integrated build system (unlike Fuego)
  - use OE/PTXdist/buildroot instead
- no integrated test runner (unlike LAVA, autotest, many others)
  - use pytest and/or custom scripts
- no test job scheduler (unlike LAVA, Fuego)
  - use Jenkins instead or use from shell
- no fixed deployment / boot process (all? others)
  - full control from client code
- library interface ⇒ not only for testing

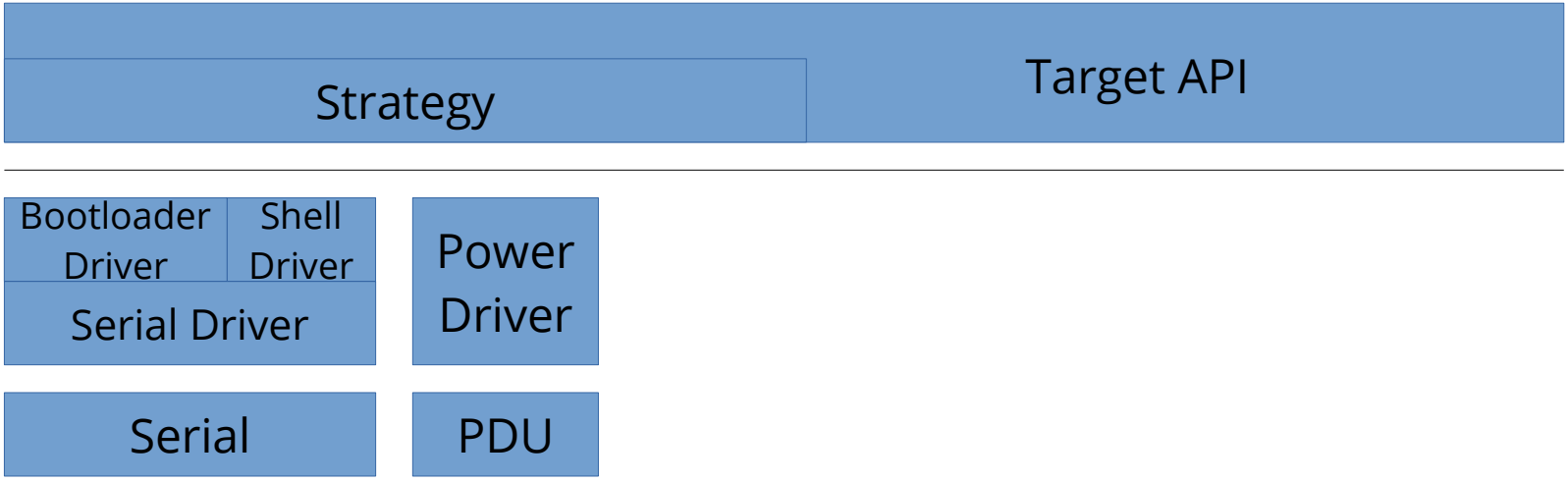
Layers


Protocol
CommandProtocol
Driver
Bootloader Driver
Shell Driver
Console Driver
/dev/ttyUSB4
Resource
Serial

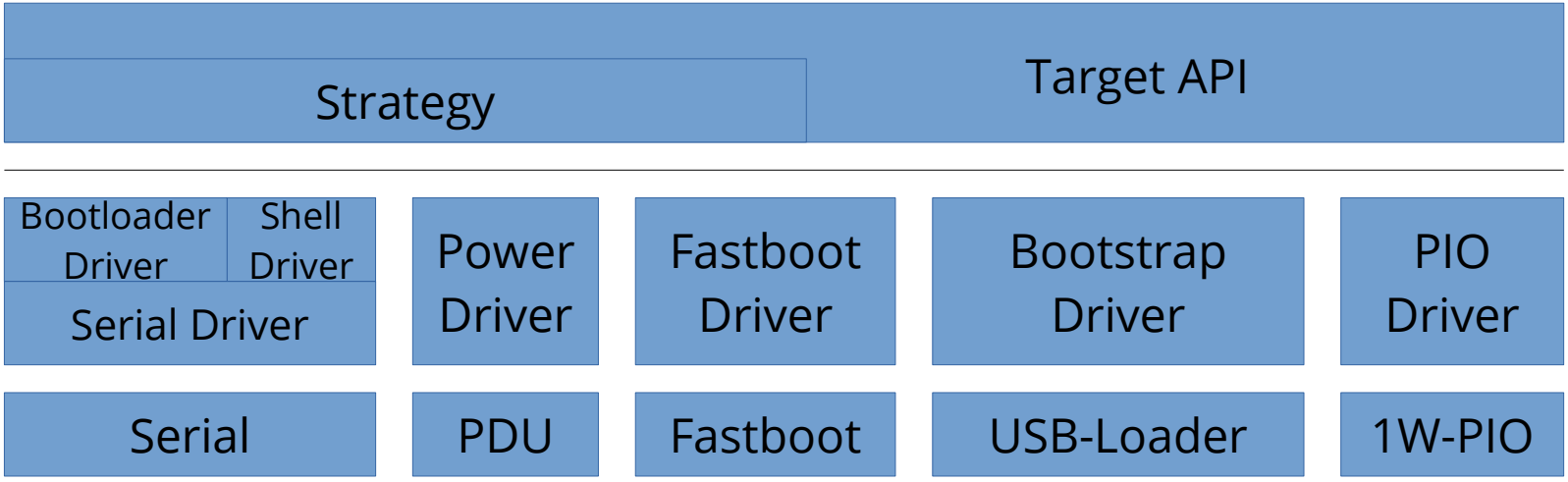
Exporters and Remote Access



Target API


Strategy
Target API
Bootloader Driver
Shell Driver
Serial Driver
Power Driver
Serial
PDU

Addtitional Interfaces



Basic Configuration

- YAML
- Describes “Targets” with
  - Resources
  - Drivers
- HW/SW-Specific parameters
- The “Environment”

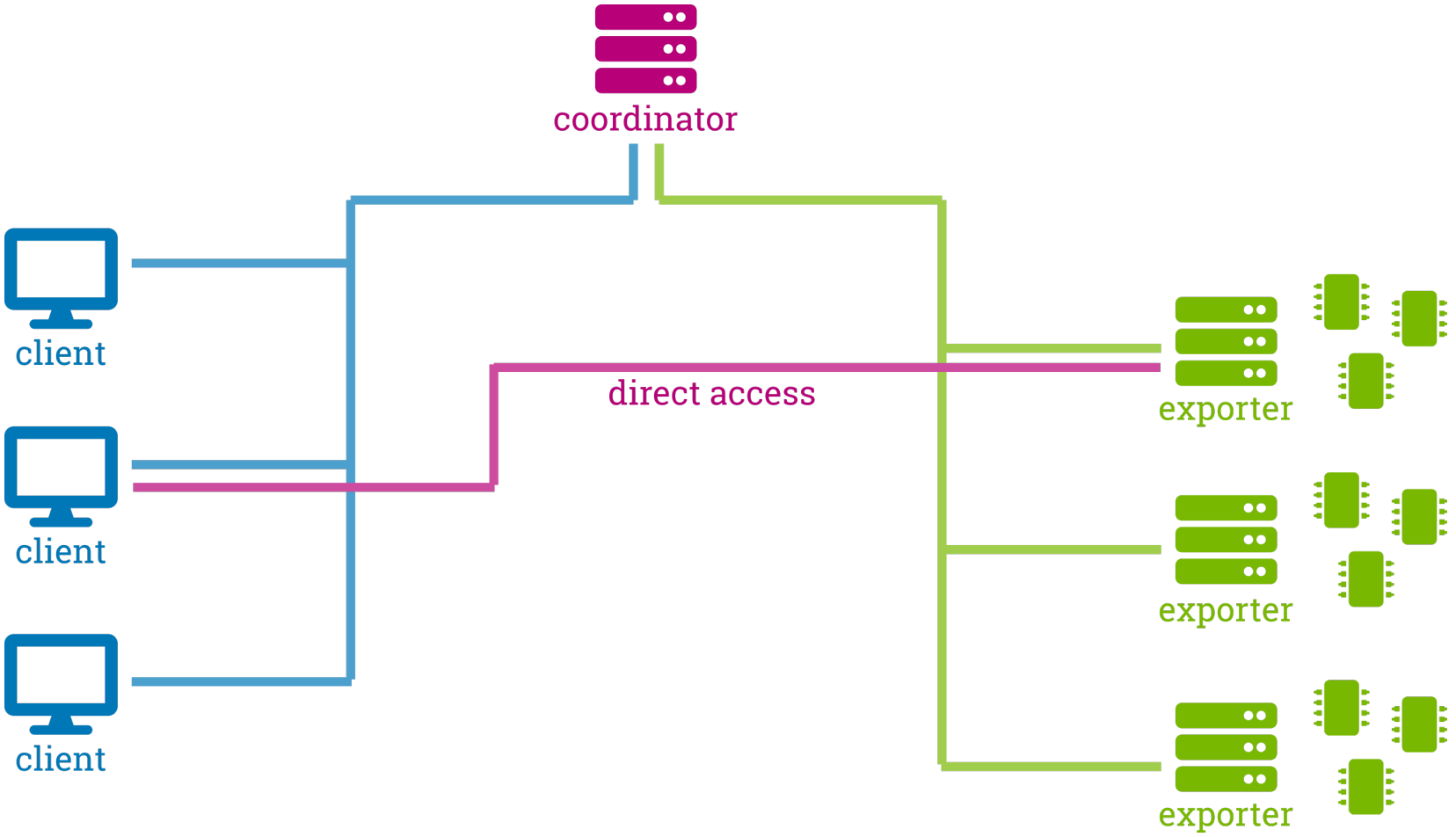
```
targets:
  main:
    resources:
      RawSerialPort:
        port: "/dev/ttyUSB0"
    drivers:
      ManualPowerDriver:
        name: "example"
      SerialDriver: {}
      BareboxDriver:
        prompt: 'barebox@[^:]+:[^ ]+ '
      ShellDriver:
        prompt: 'root@\w+:[^ ]+ '
        login_prompt: ' login: '
        username: 'root'
      BareboxStrategy: {}
```

Test Cases in pytest

- Test execution, selection and reporting is provided by pytest
- Fixtures provide access at different levels (command, strategy, target, env)
- pytest (and Python libs) make it easy to prepare test data and analyze results
- Easy to integrate in Jenkins

```
def test_hwclock_rate(command):
    """Test that the hardware clock rate is not too inaccurate."""
    result = command.run_check('hwclock -c | head -n 3')
    hw_time, sys_time, freq_offset_ppm, tick = result[-1].strip().split()
    assert abs(int(freq_offset_ppm)) < 1000
```

Remote Access



Demo: labgrid-suggest

```
=== existing device ===
  USBSerialPort for /devices/pci0000:00/0000:00:14.0/usb1/1-3/1-3.2/1-3.2:1.0/ttyUSB1/tty/ttyUSB1
  === device properies ===
  device node: /dev/ttyUSB1
  udev tags: seat, systemd, uaccess
  vendor: FTDI
  vendor (DB): Future Technology Devices International, Ltd
  model: USB_Serial_Converter
  model (DB): FT232 Serial (UART) IC
  revision: 0600
  === suggested matches ===
  USBSerialPort:
    match:
      '@ID_PATH': pci-0000:00:14.0-usb-0:3.2
  ---
  USBSerialPort:
    match:
      '@ID_SERIAL_SHORT': FT8WJPE3
  ---
```

Demo: Local Board

- environment
- interacting with the board

Demo: Remote Board

- labgrid-client, places
- lock/unlock
- power, console

Demo: Automated Provisioning

- labgrid-client -s
- using the strategy for provisioning and for bootloader shell, linux shell

Demo: pytest Test-Cases

- pytest run
- simple and more complex test case code

Demo: Reservations

- labgrid-client lock, reserve, unlock other, lock

Test Result

9 failures (+4) , 15 skipped (±0)

207 tests (+6)
Took 10 min.
add description

All Failed Tests

Test Name	Duration	Age
tests.test_userspace_services.test_wifi_regulatory_domain	2 sec	1
tests.test_userspace_services.test_ubihealthd	3.1 sec	1
tests.test_userspace_services.test_barebox_healthd	2.1 sec	1
tests.test_userspace_services.test_rfkill	16 sec	1
tests.test_linux_interfaces.test_network_interfaces	2 sec	2
tests.test_linux_interfaces.test_bluetooth_interfaces	1.9 sec	2
tests.test_linux_interfaces.test_loaded_modules	2 sec	2
tests.test_linux_ecryptfs.test_linux_ecryptfs_dep	4.1 sec	35
tests.test_linux_nvmem.test_linux_nvmem_nvstore	20 sec	108

All Tests

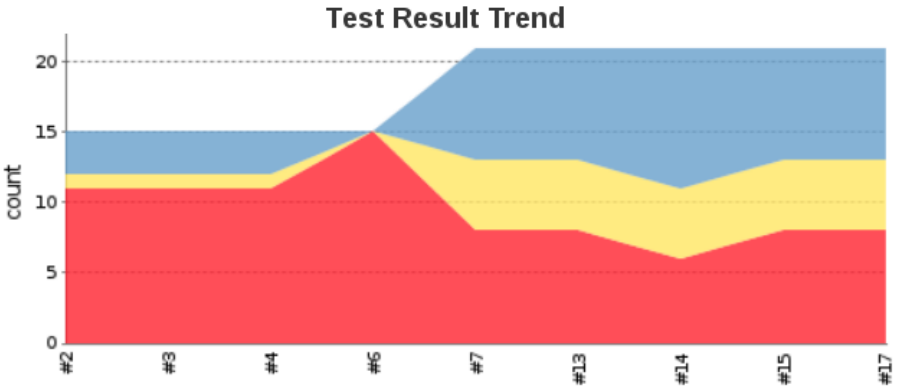
Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
tests	10 min	9	+4	15		183	+2	207	+6

Pipeline jlu/rauc/status-file

Full project name: integration-tests/combined/jlu%2Frauc%2Fstatus-file

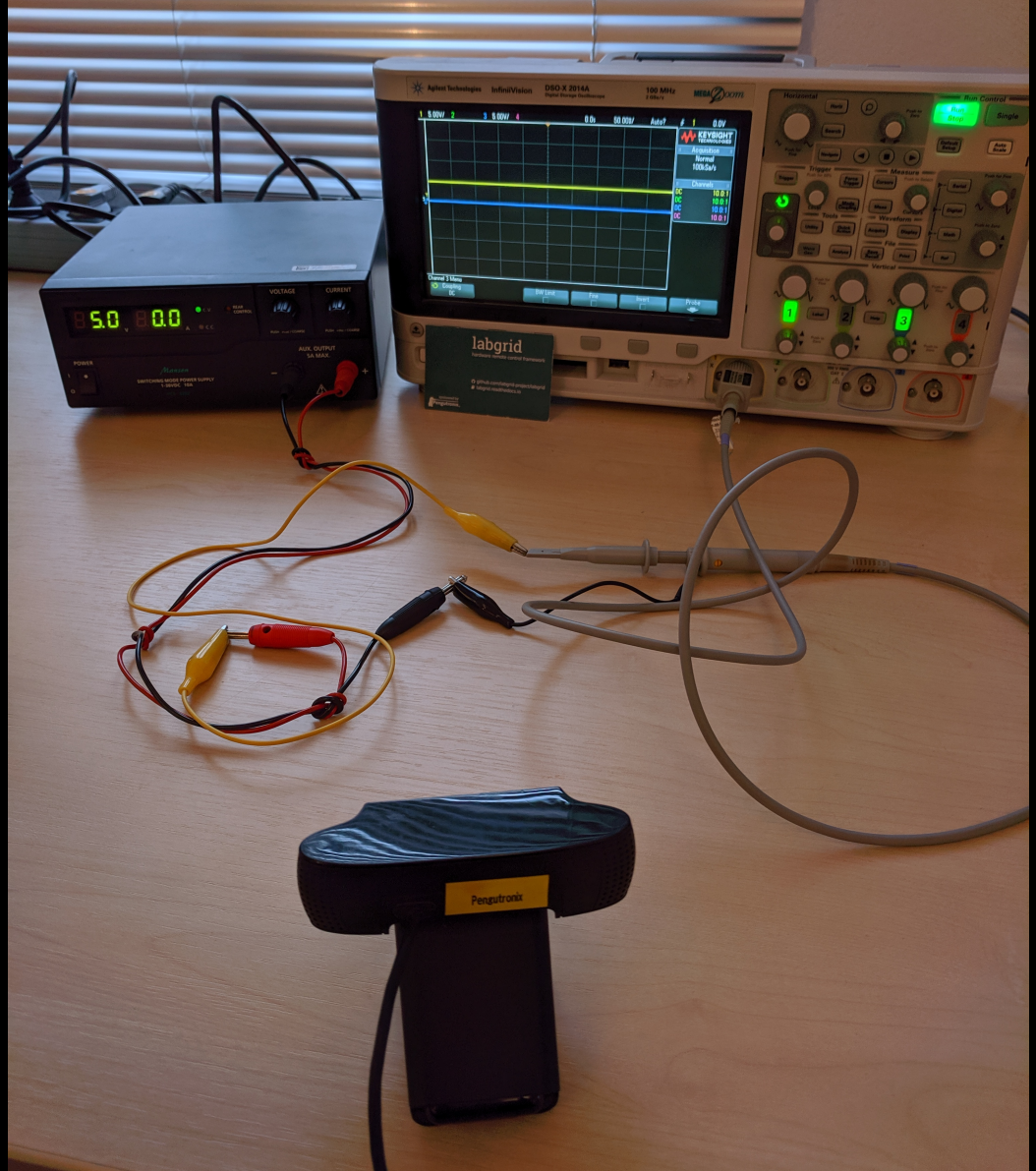
Recent Changes

Stage View



(just show failures) enlarge

	Declarative: Checkout SCM	SCM	Build ptxdist	Prepare	Build BSP	Test (pytest-barebox)	Test (pytest-shell)	Test (pytest-rauc)	Test (reason)
Average stage times:	18s	4min 42s	20s	24s	17min 23s	6s	43s	59s	45s
#17 Oct 17 17:37 — 1 commits	2s	11s	18s	32s	23min 49s	6s	1min 38s	2min 12s	2min 4s
#16 Oct 17 12:07 — 5 commits	31s	1min 15s	33s	37s failed	310ms	81ms	70ms	315ms	207ms
#15 Oct 13 — 1	981ms	34s	11s	15s	38min 4s	7s	1min 23s	2min 5s	1min 36s


5.0
0.0
VOLTAGE
CURRENT
AUX. OUTPUT
5A MAX
labgrid
Pengutronix

Demo: Other HW

- remote web-cam, PSU, scope
- SSH proxy

Questions? Discussion...

Thanks!

Scripting

```
def check_port(eth):
    command = target[ShellDriver]
    _, _, _ = command.run('arping -I {} 1.2.3.4 -c1'.format(eth))
    stdout, _, ret = command.run('ethtool -S {}'.format(eth))
    if ret:
        return False
    for line in stdout:
        ... parsing ...
        if k == 'good_frames_sent':
            if int(v) == 0:
                return False
    return True

def run_test(target):
    strategy = target[MyBareboxStrategy]
    strategy.transition('off')
    strategy.transition('shell')

    if not check_port('eth0'):
        return False
    if not check_port('eth2'):
        return False
    return True

env = Environment('myboard.yaml')
target = env.get_target()
for i in range(1000):
    if not run_test(target):
        break
```

Autoinstaller